An Integrated Approach to Finish Machining of RP Parts

Brent Stucker* and Xiuzhi Qu**

*Mechanical & Aerospace Engineering, Utah State University, Logan, UT
** Industrial & Manufacturing Engineering, University of Rhode Island, Kingston, RI

Abstract

An integrated approach to finish machining of RP parts and tools has been developed at the University of Rhode Island. Pre-processing operations, including surface offsets to add machining stock, and post-processing operations, including CNC tool-path generation, have been combined into one integrated set of software algorithms to make possible the effective finishing of near-net parts and tools from RP. An in-depth description of the uniquely developed STL vertex offset algorithm will be explored as well as an automatic method for adaptive raster milling, sharp edge contour machining and hole drilling from STL files. The time involved and surface finish benefits of the developed methodology will be compared to alternative approaches.

1 Introduction

Rapid prototyping (RP) techniques have many benefits over traditional methods for mold making. RP can create tools and parts with advanced material systems that are superior to traditional material systems, including functionally gradient structures, to optimize the tooling properties. Conformal cooling channels can be constructed within tools to reduce cycle time and increase tool productivity. And RP techniques can significantly shorten tooling creation times and reduce tooling costs when applied properly.

Rapid prototyping and manufacturing technologies are widely used in industry. These layered manufacturing approaches, decomposing the 3D CAD model into 2D layers and building the part layer by layer, have practical limitations on surface quality and accuracy. Since a layer-by-layer manufacturing method is used, a stair-step effect will occur on the part surface and, if a feature's lowest and/or highest points are not exactly on the slicing plane, inaccuracies will occur. In addition, powder-based materials are widely used in RP, such as in SLS (Selective Laser Sintering) and LENS (Laser Engineered Net Shaping) machines, and the size of particles in the material will affect the surface quality. In some cases there may be relatively large particles in the material, which exacerbate surface roughness issues. Likewise, shrinkage is a common problem for RP processes, and non-uniform shrinkage leads to dimensional inaccuracies. To date, no SFF metal fabrication process can hold sufficient tolerance for tooling. Due to these factors, CNC machining is often necessary for finishing rapid manufactured parts to an appropriate accuracy and surface finish for many applications.

The objective of this project is to develop techniques for the effective integration of CAD, RP and CNC for improving accuracy in RP-produced tool and part fabrication applications. The general methodology used in the project is shown in Fig.1. The main work focuses on the issues of the original STL file preprocessing before creating a part in an RP machine, and STL-based automatic tool path generation for finish machining after part creation.

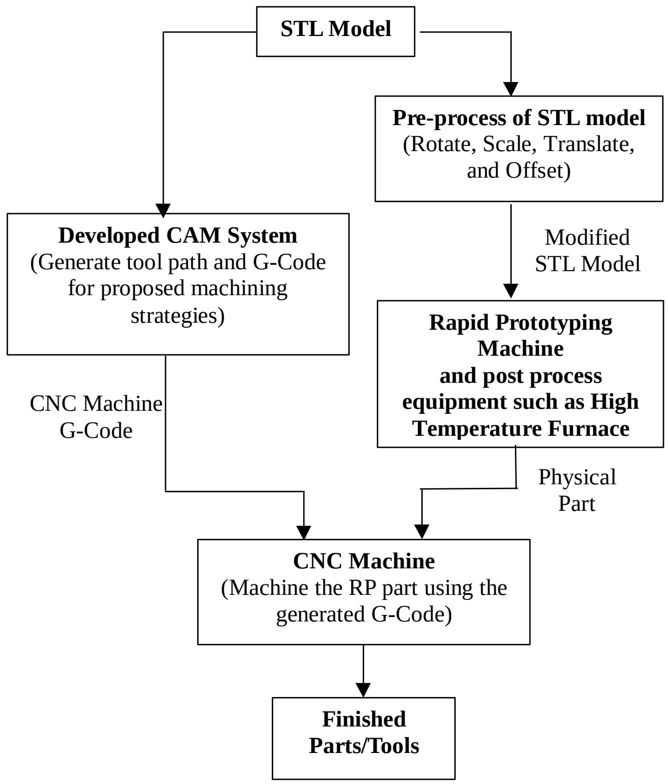


Figure 1. Flowchart of the whole processing approach

2 Preprocessing of STL File

The accuracy and surface finish of parts manufactured by RP are material and process dependent. In order to obtain high quality parts or tools, some pre-processing of the 3D model is necessary prior to its RP fabrication. Translation and rotation operations are used to optimize the part position and orientation. Scaling of the model is used to compensate for overall shrinkage during the process, including post-processing steps. A 3D offset method is used to add machining stock to the original model to ensure that the part is “steel-safe.” (A steel-safe part is one where enough extra material is present, in all dimensions and on all features, to allow finish machining to the desired accuracy and surface finish). Adding machining stock is often a time-consuming, labor-intensive process. This paper presents an effective, automatic offsetting algorithm developed specifically to enable the addition of machining stock to STL files that will be produced using RP.

2.1 Translate, rotate and scale

The implementation algorithms for these basic operations simply translate, rotate or scale each triangle vertex of the 3D model, and then model information is recalculated.

For rotate operations, an additional calculation is needed to modify the unit normal values of the triangles.

Suppose that $x_{i,j}$, $y_{i,j}$ and $z_{i,j}$ represent the *x*, *y* and *z* coordinate of *j*th vertex of *i*th triangle in the original model, and $x^*_{i,j}$, $y^*_{i,j}$ and $z^*_{i,j}$ are their new coordinates after transforming, then the equations used to make these transformations are,

- **Translate**

$$\begin{aligned} x^*_{i,j} &= x_{i,j} + x_{Translate} \\ y^*_{i,j} &= y_{i,j} + y_{Translate} \\ z^*_{i,j} &= z_{i,j} + z_{Translate} \end{aligned} \tag{1}$$

Where $x_{Translate}$, $y_{Translate}$ and $z_{Translate}$ are the required translating value along *x*, *y* and *z* axis.

- **Scale**

$$\begin{aligned} x^*_{i,j} &= x_{i,j} * x_{Scale} \\ y^*_{i,j} &= y_{i,j} * y_{Scale} \\ z^*_{i,j} &= z_{i,j} * z_{Scale} \end{aligned} \tag{2}$$

Where x_{Scale}, y_{Scale} and z_{Scale} are the given scale factors along *x*, *y* and *z* axis.

- **Rotate**

Suppose the rotating sequence is first about axis *x*, then *y* and finally *z*, then

$$\begin{aligned} \begin{bmatrix} x^*_{i,j} \\ y^*_{i,j} \\ z^*_{i,j} \end{bmatrix} &= \begin{bmatrix} \cos\alpha_z & -\sin\alpha_z & 0 \\ \sin\alpha_z & \cos\alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} \cos\alpha_y & 0 & \sin\alpha_y \\ 0 & 1 & 0 \\ -\sin\alpha_y & 0 & \cos\alpha_y \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_x & -\sin\alpha_x \\ 0 & \sin\alpha_x & \cos\alpha_x \end{bmatrix} \bullet \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{bmatrix} = \\ &\begin{bmatrix} \cos\alpha_z\cos\alpha_y & -\sin\alpha_z\cos\alpha_x + \cos\alpha_z\sin\alpha_y\sin\alpha_x & \sin\alpha_z\sin\alpha_x + \cos\alpha_z\sin\alpha_y\cos\alpha_x \\ \sin\alpha_z\cos\alpha_y & \cos\alpha_z\cos\alpha_x + \sin\alpha_z\sin\alpha_y\sin\alpha_x & -\cos\alpha_z\sin\alpha_x + \sin\alpha_z\sin\alpha_y\cos\alpha_x \\ -\sin\alpha_y & \cos\alpha_y\sin\alpha_x & \cos\alpha_y\cos\alpha_x \end{bmatrix} \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{bmatrix} \end{aligned} \tag{3}$$

Where α_x, α_y and α_z are the rotating angles about *x*, *y* and *z* axis.

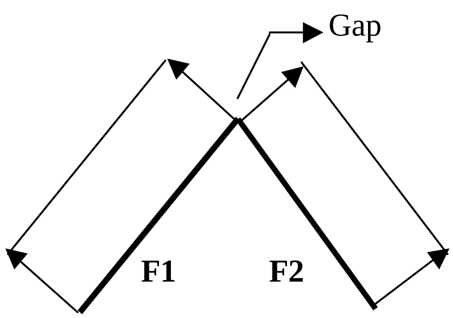
2.2 3D Model Offset

In order to make sure there is enough material left on the surface to be machined, adding "skin" to the original model is necessary. From experiments performed in the Rapid Manufacturing Center (RMC) at the University of Rhode Island (URI)[6], it was shown that the shrinkage level varied, even for the same SLS and furnace parameter settings. This variation also requires offsetting of the original model to guarantee that even the features with the largest shrinkage levels still have material left for machining.

Offset techniques for curves and surfaces have been extensively studied, and survey papers have been written by Pham [1] and Takashi [2]. Many exact and approximation methods have been developed to generate offset curves and surfaces for different types of curves and surfaces. The specific problem for this project was to generate offsets for 3D solid models, which involves not only the geometrical issue of offsetting each individual surface in the model but also the topological issue of reconnecting these offset surfaces into a closed 3D model. Generally, offsets of 3D models are achieved by first offsetting all surfaces of the model and then trimming or extending these offset surfaces[3][4][5].

2.2.1 Conventional offset method

To offset a 3D model in the STL format, the most direct method would be to offset each triangular facet with the given offset distance in its corresponding normal direction. However, this will result in intersections or gaps between the offset surfaces of two neighboring triangles. As shown in Fig.2a, a gap is formed between two offset surfaces F1 and F2 when the angle between them is convex. Conversely, an intersection or overlap occurs between offset surfaces, as shown in Fig.2b, when the angle between them is concave. In order to make closed 3D models from these triangular offset surfaces, it is necessary to identify all of the intersections, and then trim the surfaces on the line of intersection, and to identify all of the gaps and extend the surfaces to fill them. This can be quite complex, since thousands or millions of triangular facets may exist when representing complex 3D models using the STL format.



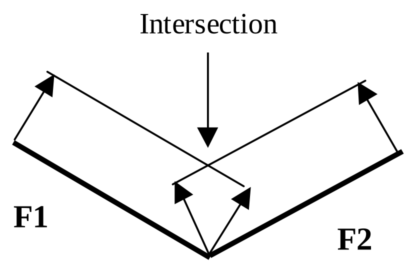


Figure 2a. Gap between offset surfaces   Figure 2b. Intersection between offset surfaces

2.2.2 Vertex-based offset method

This problem can be avoided if the vertices, instead of the triangular facets, are offset. As shown in Fig.3, when offsetting the vertices the relationship between facets will remain and there is no need to recalculate the triangle intersections. The challenge when utilizing this method is how to effectively calculate the offset vector for each vertex, taking into accounts the offset direction and magnitude for all of its surrounding triangular facets.

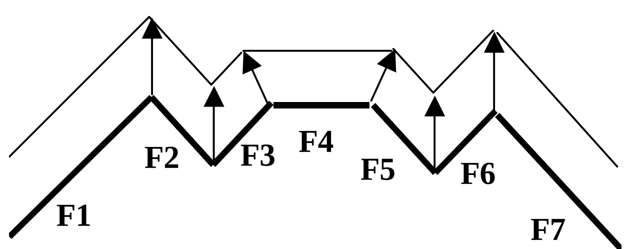


Figure 3. Offsetting vertices of a 3D model

This paper presents a new method for calculating the offset vector for a vertex by using the weighted sum of the normal vectors of the triangular facets connected to the vertex as follows:

$$\vec{V}_{Offset} = \sum_{j=1}^{n} W_j * \vec{N}_{i,j} \tag{4}$$

where W_j are the weighting coefficients associated with each triangular facet. Before presenting the method for calculating the coefficients W_j, the relationship between $\vec{V}_{Offset}$ and $\vec{N}_{i,j}$ will be explored. As shown in Fig. 4, $P_{i,original}$ is the vertex position before offset, and $P_{i,new}$ is the position after offset. Given the offset vector $\vec{V}_{Offset}$ and offset distance d_{Offset}, the following equation can be derived,

$$P_{i,new} = P_{i,original} + \vec{V}_{Offset} * d_{Offset} \tag{5}$$

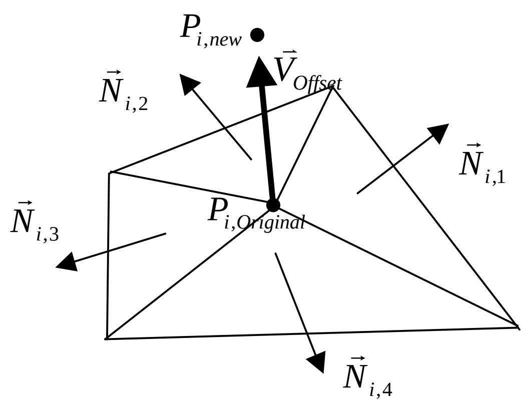


Figure 4. Illustration of the relationship between $\vec{V}_{Offset}$ and $\vec{N}_{i,j}$

From geometrical information, it is known that the perpendicular distance from $P_{i,new}$ to any original connecting triangular surface should be exactly the offset distance d_{Offset} since $P_{i,new}$ is located on the intersection point of the offset surfaces of all of the connecting triangular facets. That means that the following relational equation should be met,

$$(P_{i,new} - P_{i,original}) \bullet \vec{N}_{i,j} = d_{Offset} \tag{6}$$

From equations (5) and (6), the following equation is derived.

$$\vec{V}_{Offset} \bullet \vec{N}_{i,j} = 1 \tag{7}$$

Substituting for $\vec{V}_{Offset}$ results in the following equation.

$$(\sum_{j=1}^{n} W_j * \vec{N}_{i,j}) \bullet \vec{N}_{i,k} = 1 \quad (k = 1 \text{ to } n) \tag{8}$$

By solving Eqn. (8), the weighting coefficients, W_j, can be calculated, and then offset vectors can be caluculated using Eqn. (4).

After the offset vector $\vec{V}_{Offset}$ for each unique vertex is found, the new position of each vertex can be calculated using Eqn. (5). Depending on whether an outward or inward offset is desired, d_{Offset} can be either positive or negative respectively. All of the new position values for each unique vertex are then used to construct a new set of triangular facets, which represent the desired offset model in the STL format.

3 Finish Machining Strategy

Machining strategy directly influences both machining quality and machining efficiency. When considering both accuracy and machining efficiency, adaptive raster milling of the surface, plus hole drilling and sharp edge contour machining is posited as the best combination of machining strategies. No commercial software packages capable of automatically generating tool paths from STL 3D models combine these machining strategies. Therefore, several integrated algorithms were developed as part of this project, including sharp edge detection and drilled hole recognition algorithms and tool path generation routines for raster milling, sharp edge contouring and hole drilling.

3.1 Adaptive Raster Milling

When raster machining is used for milling operations, stepover distance is the parameter that controls machining accuracy and surface finish. For higher accuracies and surface finishes, a smaller stepover distance is required. Often the cusp height of material left after the model is machined is used as a measurement of the surface quality.

Through analysis, it is found that the relationship between stepover distance d, cusp height h, cutter radius r, the surface normal $\vec{N}_{Triangle}$ of facet to be machined, and the unit vector $\vec{N}_{Stepover}$ along the stepover direction, can be described as follow:

$$d = 2.0 * \sqrt{h(2r-h) * (1-(N_{Triangle} \bullet N_{Stepover})^2)} \tag{9}$$

When machining the model, the cutter radius and milling direction are the same for all triangle surfaces. If given the required cusp height h, d is only related to the triangle normal vector. For surfaces with different normal vectors, the stepover distance obtained from Eqn. (9) will be different for each surface. If a constant stepover distance is used, the minimum calculated d should be chosen as the stepover distance for milling the entire part, in order to guarantee a maximum cusp height overall.

Smaller stepover distances, however, will lead to longer programs and machining times. Therefore, an adaptive stepover distance for milling operations according to local geometry has been developed to enable both accuracy and machining efficiency. This means that stepover distances are calculated dynamically for each tool pass independently, using the allowable maximum cusp height to determine the stepover distance for the next tool pass. When using an adaptive strategy, the local minimum stepover distance is typically bigger than the overall minimum value for the part, thus each stepover distance is no smaller than necessary to maintain a maximum cusp height for any particular location on the part. This dramatically improves machining efficiency.

For tool path generation using raster milling, the often-used method for generating cutter paths is to first calculate the CC (cutter contact) point on the model, then use the normal vector direction and cutter size information to derive the CL (cutter location) point. However, this method typically requires additional gouge detection algorithms, which is normally a complicated process, to adjust cutter location in order to avoid overcuts for some complex surfaces. Another serious problem in metal cutting is that the tool paths produced by this method are not planar, while the CC points, which are generated by slicing the part surface with a plane, are planar [7]. An alternative, offset surface method can directly calculate the cutter location points and generate gouge-free parallel tool paths [8][9]. Since offset algorithms were developed for the above-mentioned skin addition operation, these same algorithms can be utilized for tool path generation. The adaptive raster milling tool paths are generated by first creating an offset surface with the offset distance equal to the cutter radius, then slicing the offset model using the adaptive tool-path technique and finally identifying the top contour envelope so that only the contours representing upward-facing features are machined.

3.2 Sharp Edge Contour Machining

Sharp edges are the intersection curves between features and surfaces. These edges are typically used to define critical dimensions and often need to be machined accurately and smoothly. However, when using raster milling alone, edges parallel to the milling direction may be missed and cannot be machined accurately even when specifying a small cusp height, resulting in large errors.

For complicated edges not parallel to the milling direction, raster milling is ineffective for creating smooth edges. To make smooth edges, very small stepover distances would be required, making adaptive tool path generation less efficient. Thus a method of milling along recognized sharp edges was developed to solve this problem.

Contour machining is a popular strategy for machining parts. To implement it using STL models is not trivial, as all the feature information is lost when the STL file is created. The main characteristics of the developed sharp edge contouring algorithms are to automatically identify all of the contour edges that should be machined using generic recognition routines, unlike the traditional way of specifying these edges by user input, and then generate gouge-free tool paths using the 3D offset model.

3.3 Hole drilling

Circular holes are common features in parts and tools. Using milling tools to create holes is inefficient and the circularity of the holes is poor. Therefore, a machining strategy of drilling holes using the correct-sized drilling tool was developed.

The most challenging aspect is to recognize drilled holes automatically, regardless of hole orientation with respect to the z axis. In the STL data format, 3D geometry is represented by a collection of unordered triangular planar facets. Thus all feature information is lost. The intersection curve between a hole and a surface is typically a closed loop. By using this information, a hole recognition algorithm begins by identifying all closed loops made up of sharp edges from the model. These closed loops may not necessarily be the intersection curves between holes and a surface, so a series of hole checking rules are used to remove the loops that do not correspond to drilled holes. The remaining loops and their surface normal vectors are used to determine the diameter, axis orientation, and depth for drilling. From this information, tool paths can be generated.

4 Experiment and Results

The benchmark part shown in Fig. 5 was designed to verify the developed machining strategies and automatic tool-path generation algorithms. This part includes a variety of features, such as different types of surfaces, complicated sharp edges and holes in different orientations.

In order to compare the results using different finish machining strategies, two parts, called part1 and part2, were first built from a Copper Polyamide material on a DTM Sinterstation 2500*plus* machine. Before running on the Sinterstation machine, the 3D CAD file was pre-processed. Based on previous experience, the parts were scaled with a factor of 1.025 in the x and y directions and 1.015 in the z direction. A value of 0.035" was used as a 3D offsetting value for the whole model. After building the parts, 22 different dimensions were measured for each part, and the results are shown in Fig.6.

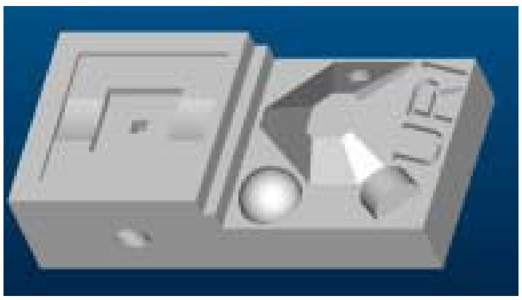


Figure 5. Designed Benchmark

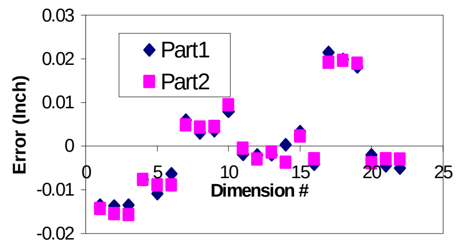


Figure 6. Errors after SLS Processing

The parts were then CNC finish machined using a 3-axis Bridgeport CNC milling machine. Part1 was machined using the strategy presented in this paper: adaptive stepover distance raster milling of the surface plus hole drilling and sharp edge contour machining. Part2 was machined using raster milling with a constant stepover distance.

Comparison results are given in Fig. 7 and Table 1. As can be seen, part1 was finished in less time, and its surface smoothness and dimensional accuracy were better than that of part2.

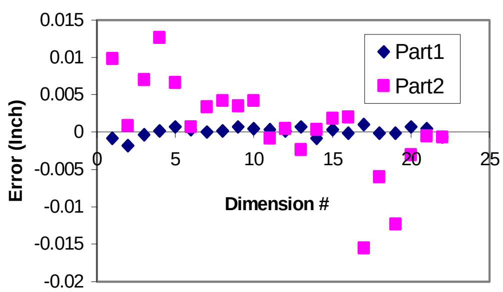


Figure 7. Errors after Machining

	Machining Time(Min.)	Maximum Cusp Height Approx. (Inch)	Smoothness of Vertical Surface
Part1	137	0.0005	High
Part2	175	0.0015	Middle

Table 1. Comparison Result

This study shows an initial applicability of the presented procedure and the improved capabilities of this machining strategy for finish machining of rapid manufactured parts. The 3D offset algorithm, sharp edge and hole feature recognition algorithms and tool path generation methods were successfully verified.

5 Conclusion

This paper presents a comprehensive approach for the finish machining of RP parts. Pre-processing operations, including surface offsets to add machining stock, and post-processing operations, including CNC tool-path generation, have been combined into one integrated set of software algorithms to make possible the effective finishing of near-net parts and tools from RP. A unique offset algorithm was developed to automatically add “skin” to the original STL file to ensure a steel-safe part is left for finish machining after the rapid manufacturing process. The machining strategy of raster milling of the surface plus hole drilling and sharp edge contour machining has been presented and their corresponding automatic tool path generation algorithms were developed. Fully automatic 3D model offsets and optimized, automatic tool-path generation based on STL files should minimize the man-hours and milling times associated with finish machining of RP parts and tools..

References

[1] Pham, B., “Offset curves and surfaces: a brief survey”, Computer-aided Design, 24(4), pp.223-229, 1992

[2] Maekawa, T., “An overview of offset curves ad surfaces”, Computer-Aided Design, 31(3), pp.165-173, 1999

[3] Farouki, R. T., “Exact offset procedures for simple solids”, Computer-Aided Design, 2(4), pp.257-279, 1985

[4] Satoh, T., Chiyokura, H., “Boolean operations on sets using surface data”, ACM SIGGRAPH: Syposium on solid modeling foundations and CAD/Cam applications, Austin, Texas, USA, pp.119-127, 1991

[5] Forsyth, M., “Shelling and offsetting bodies”, Proceedings of Third Symposium on Solid Modeling and Applications, Salt Lake City, Utah, USA, pp.373-381, 1995

[6] Malhotra, M., “Rapid Manufacture of Zirconium Diboride/Copper & Dimensional Stability of RapidSteel® 2.0”, MS thesis, University of Rhode Island, RI, 2000

[7] Hwang, J.S., "Interference-free too-path generation in the NC machining of parametric compound surfaces", Computer-Aided Design, 24(12), pp.667-676, 1992

[8] Kim, K.I., Kim, K. , "A new machine strategy for sculptured surfaces using offset surface", International Journal of Production Research, 33(6), pp.1683-1697,1995

[9] Choi, B.K., Kim, D.H., Jerard, R.B., "C-space approach to tool-path generation for die and mould machining", Computer-Aided Design, 29(9), pp.657-669, 1997

Acknowledgements:

This work was performed at the Rapid Manufacturing Center and was fully funded by its member companies. Member companies who helped fund this work included: Gillette Corporation, Infinite Group, Optomec Design Company and 3D Systems (formerly DTM Corporation).